

Vehicle and Person Tracking in UAV Videos

Jiangjian Xiao, Changjiang Yang, Feng Han, and Hui Cheng

Sarnoff Corporation

{jxiao, cyang, fhan, hcheng}@sarnoff.com

Abstract. This paper presents two tracking approaches from Sarnoff Corporation to detect moving vehicles and person in the videos taken from Unmanned Aerial Vehicles (UAV). In the first approach, we combine layer segmentation approach with background stabilization and post track refinement to reliably detect small moving objects at the relatively low processing speed. Our second approach employ a fast tracking algorithm that has been optimized for real-time application. To classify vehicle and person from the detected objects, a HOG based vehicle v.s. person classifier is designed and integrated with the tracking post-processing. Finally, we report the results of our algorithms on a large scale data set collected from VIVID program and the scores evaluated by NIST CLEAR program.

1 Introduction

Object tracking is a classic research topic in computer vision and has been investigated by computer vision researchers for a long time[1–5]. With the availability of the low cost video cameras, a huge amount video data is recorded every day to be analyzed. Therefore, a reliable and automated video content analysis process is very important. In such process, object tracking is the fundamental block for the high level content analysis and exploitation. Especially, in the intelligence community, UAV video has been one of the fastest growing data sources and it has been extensively used in intelligence, surveillance, reconnaissance, tactical and security applications[5]. Based on the analysis of the routinely captured video from UAV, a series of intelligent content services for intelligence community are provided to greatly improve the analyst’s capabilities in motion imagery exploitation and to enhance security against terrorist attack at US installations at home and abroad.

Traditionally, the object tracking includes two parts: moving object detection and tracking the detected object over the frames. The first part is to detect the interesting moving object based on the motion information such as optical flow or background subtraction. The second part is to maintain a consistent identity on the object based on the appearance, shape, or kinematic information when the object is either moving or becomes stationary over the frames. In our CLEAR evaluation task, an additional sub-task is required to classify the moving objects into two categories: vehicle and person. Therefore, an appearance based object classifier is needed to be designed for this task.

In this paper, we present two tracking approaches and conduct performance evaluation between these two approaches on a large UAV video data set. Our first approach

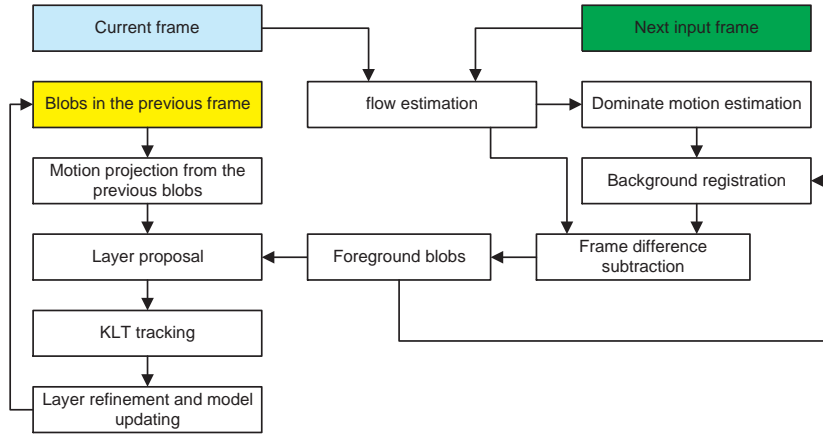


Fig. 1. The flow chart of layer-based tracking algorithm. At each time instance, the inputs include two frames and the previous blob set in the previous frame, and the output is the updated blob set with ID information.

is layer-based tracker, where the background and foreground moving objects are represented by different layers respectively. During the detection and tracking process, each layer has maintained an independent appearance and motion model. After performing the tracking process on the video, we also apply a post-tracking refinement process to link the track fragments into a long consistent track ID to further reduce false alarm and increase detection rate. In our second tracking algorithm, SNIFF, we optimize the tracking code and speed up the algorithm for real time applications. This algorithm is based on robust change detection and optical flow based linkage[3]. This algorithm has been successfully deployed on a series of real products in Sarnoff Corporation.

The remainder of this paper is organized as follows. Section 2 discusses the layer based tracking approach and post-tracking refinement. Section 3 presents the fast tracking algorithm broadly used in Sarnoff products. In section 4, a HOG based vehicle v.s. person classifier is discussed. The experimental results and evaluation scores for both vehicle and person tracking are reported at Section 5.

2 Layer-based tracker and post linking

In our layer-based tracking algorithm, we combine background stabilization and layer representation to detect and track the moving objects. Compared with the ground stationary or PTZ cameras, tracking in UAV video is more challenging due to the significant camera motion and parallax affect[6, 7]. In order to reduce the camera motion effect, a frame-to-frame video registration or stabilization process is necessary. In this step, the frames are registered through the background regions excluding the foreground moving objects. However, the video registration may often fail when the camera has a fast motion or strong illumination change. In such case, the detection and tracking pro-

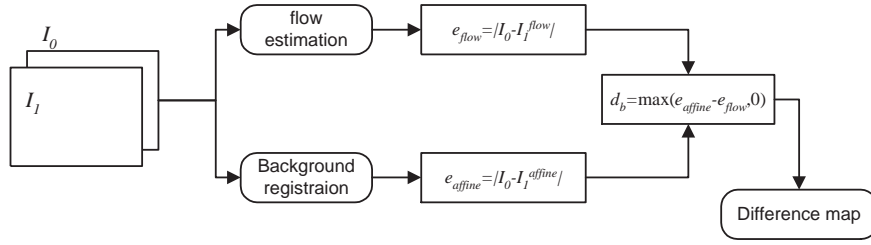


Fig. 2. The process of difference subtraction between two image residues, where I_1^{flow} and I_1^{affine} are the warping images of frame I_1 respectively. The resulted difference map will be used for the foreground blob detection.

cess should be temporarily suspended and a motion prediction will fully take control until the reliable registration becomes available.

2.1 Layer proposal generation

Figure 1 shows the flowchart of our layer-based tracking algorithm. In the approach, we first estimate dense optical flow between the new coming video frame and the current reference frame. Then, based on the dominant motion extracted from the optical flow, a global registration is performed to align the background region. During the alignment, the outlier regions are detected and segmented out from the images by the following subtraction process[9]. Therefore, the registration becomes more robust and the small motion of the moving object can be possibly detected.

Instead of using simple subtraction between the aligned frame, we propose a difference subtraction method to reduce motion noise and compensate the illustration change such as automatic gain control. In our approach, two warping images, I_1^{flow} and I_1^{affine} , are first computed from affine warping and flow warping as shown in Figure 2. Next, the corresponding residual images, e_{flow} and e_{affine} , are obtained by subtraction between I_0 with these two warping images. Since the optical flow approach attempts to minimize image residue e_{flow} between two input frames, the image residue for independent moving object will be also reduced by the minimization process and generally is smaller than those in e_{affine} . For those occlusion regions, the image residue, e_{flow} , is similar to e_{affine} , since in both cases there are no exact correspondences to minimize the image residues. Hence, if we subtract the two image residues, e_{affine} and e_{flow} , a second order residual difference, d_b , are generated such that

$$d_b = \max(e_{affine} - e_{flow}, 0). \quad (1)$$

In this new residual difference map, the independent moving object will have strong response, and image residues at the occlusion regions will be canceled. This approach can also effectively cancel the illumination inconsistency due to camera motion since both residual images has same response for illumination variations. After applying our difference subtraction approach, the image residues due to occlusion regions are filtered out and only the independent moving object has strong response.

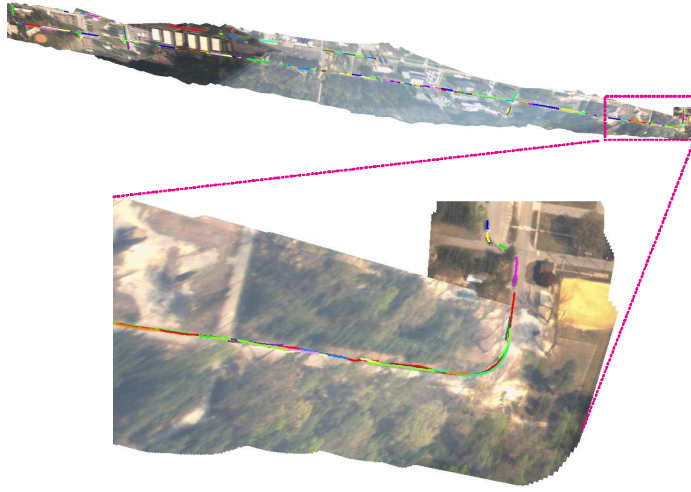


Fig. 3. Mosaic of one video sequence. The detected tracks are also overlapped on the mosaic. The bottom is the zoom view of partial mosaic.

After the difference subtraction, the foreground blob can be detected by either simple threshold or graph cut segmentation. In our approach, we apply the more sophisticated graph cut method to segment the foreground object from the background[9]. After that, a merging process will merge the small blob fragments into a reasonable size based on the motion similarity and in-between distance. Therefore, an initial blob set, S_1 , is obtained. Then, from the previous frame, we project the previous blob set, S_0 , by using its corresponding motion parameters into this frame to create a warped blob set, \tilde{S}_0 . Once two blob sets are obtained, a maximal likelihood association process is applied to associate the blobs and reassign the pixels into the blob set. For remaining un-associated blobs, a new ID will be assigned. As a result, the set of moving targets is detected and the corresponding layer proposal is generated.

2.2 Layer refinement and model update

After the above mentioned motion target indication step, the blob shape of the moving object may not be accurate enough due to occlusion or textureless. For example, background region may be mis-assigned into the blob, or some portion of the moving object may not be fully detected in the blob. Therefore, we need to refine the blob shape to exclude the background pixel from the blob and also include the nearby undetected object pixels into the blob as many as possible.

In our approach, we first apply a KLT tracking process to re-estimate the motion model of the blob[2, 10]. In KLT tracker, a motion model is approximately represented by an affine transformation, such that,

$$I_1(Ax + t) = I_0(x), \quad (2)$$



Fig. 4. Post linking result. Top: The ID of blue car has been changed several times due to occlusion by trees. Bottom: The car’s ID is maintained after linking process.

where A is a 2D matrix and t is the translation vector. This equation can be iteratively solved by gradient descent method for the pixels within the blob. Similarly, a residual difference map, d_f , is generated as Eq . Then, we apply a graph cut approach by using d_f against d_b to identify the nearby supporting pixels and exclude background outliers from the blob[8]. In the graph cut algorithm, a smoothness energy term is introduced to naturally solve the problem. This smoothness energy term fully exploit the similarity information around the neighboring pixels, and constrain the partition more prone to high gradient boundary that is more likely corresponding to object boundary.

2.3 Post-linking process

In order to further reduce false alarm and increase detection rate, we also propose a post-linking process to link the track fragments into a smooth long track with a same identity. In our approach, we mainly employ track kinematic property to evaluate the association probability between different tracks. Our approach includes three main steps. In the first step, we create a mosaic from the input video sequence and re-project the moving object tracks into the mosaic space. Figure 3 shows the mosaic of one video sequence. Next, we smooth the track fragment by using Kalman filter and estimate the speed at each time instance. If the track is missing at a certain frames, an interpolation process will be applied to recover the location and speed information for these frames. After that, an extrapolation process is applied for each smoothed track. Finally, a probability association metric is applied for the tracks which are temporal closing or overlapped. In this step, a greedy algorithm is designed to minimize the cost function by the near-neighbor association. Figure 4 shows one result before and after the linking process. With the post-linking process, some missing tracks are recovered and track fragments are dramatically reduced for most cases. Therefore, both the number of ID switching and false alarm rate are reduced to a satisfactory low level.

3 SNIFF object tracking algorithm

In this section, we will discuss another tracking algorithm, which was developed at Sarnoff Corporation for real-time application[3, 11]. The algorithm is based on robust

change detection and optical flow based linkage as shown in Figure 5. At the first stage, a robust change detection is used to find the moving object position every frame or every other fixed number of frames (depending the program configuration). At the second stage, optical flow technique is used to find the association between the detected objects in the frames.

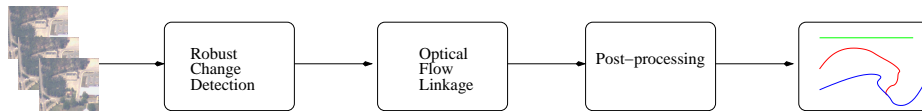


Fig. 5. Flow chart of the Sniff algorithm

At the change detection stage, the first step is to align the consecutive frames then to subtract the aligned images. In Sniff algorithm, the default image alignment is performed using the direct method or flow-based method. There is another image alignment method, called feature-based method. The benefit of direct method is that it is fast with the help of using pyramids. The drawbacks are that it is sensitive to the large illumination changes, occlusions, low textured surface, or rapid motion. The feature based method such as Harris corner detection can effectively deal with such difficulties. The Sniff algorithm provides both alignment options for different requirements.

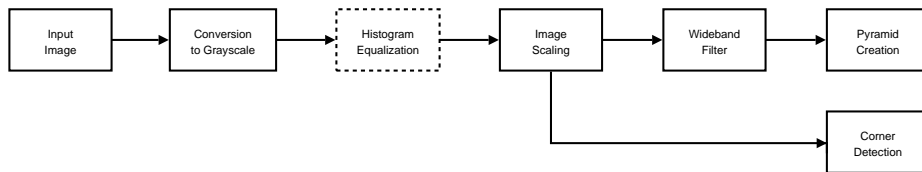


Fig. 6. The pre-processing stage for Sniff tracking. The images are first converted to grayscale format. Then an optional histogram equalization step is applied. A Gaussian pyramid is built for the optical flow computation. Harris corner detection is another option for image alignment in the later stage.

The simple image subtraction is sensitive to the sudden illumination changes, and the vegetable-textured areas. So the Sniff algorithm chooses normal flow rather than simple image subtraction. The normal flow will generate object masks in the image. Due to noise and/or occlusions and shadows, the masks might be broken. Morphological operation such as dilation is applied to grow the masks back to the original size. In the real implementation, the Sniff algorithm steps up to one level of the image pyramid then steps down one level to simulate the dilation procedure.

At the association stage, the optical flow is computed between consecutive frames. The correspondences between the pixels in the masks of different frames is associated

using the optical flow. To speed up this association processing, every four pixels are glued together to form a super-pixel. The Sniff also performs an AND operation on the associated multi-frame masks to remove the false alarms.

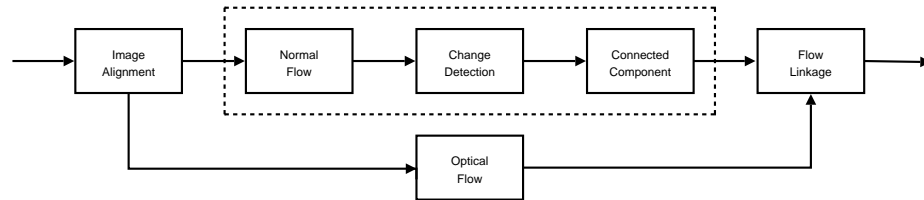


Fig. 7. The Sniff tracking flow chart. Once the images are aligned, the normal flow based change detection is applied to the aligned images. Then morphological operation is used to fill the holes and the broken parts. The connected component algorithm is used to find the blobs in the images. An optical flow method is used to link the blobs in the different frames and assign the ID accordingly.

3.1 Post-processing: use of global tracking trajectory

One of the biggest difficulties in Sniff algorithm is that it does not handle slowly-moving objects, static objects, and occlusion well, since any broken one frame within the linkage will lose track of the objects. To deal with this problem, the algorithm book-keeps the lost objects for a fix period of time. Later on, if the lost objects re-appear, then the lost objects will be re-associated with the current track based on the size of the object and the central moments of the flow vectors.

Even though the Sniff alleviated the above presented difficulties, it cannot handle the problems due to a long period time of occlusion or slow moving objects. The reason behind is that it is a local method and the global tracking information is ignored. However such global tracking information is vital for object tracking. For instance, human vision system utilizes such global information to find the moving objects, even very small one.

To improve the algorithm, we first build a mosaic of tracking trajectory for exploit global information. However, in the 2D mosaic space, the tracking trajectories may be very ambiguous, since a lot of trajectories are overlapped as shown in Figure 3. We then lift up the trajectories into 3D spatio-temporal space. Such lifting makes the trajectories more distinct even they are overlapped in 2D space. The next stage is to connect the fragmented trajectories in 3D space based on the global proximity, continuation, and collinearity. Usually when the occlusion happens, the position and size of the objects before and after occlusion is not accurate, which makes the association method based on local information such as Kalman filter ineffective. On the other hand, global linkage will overcome such shortcoming using global information, which makes the algorithm more robust, even for very large occlusion.

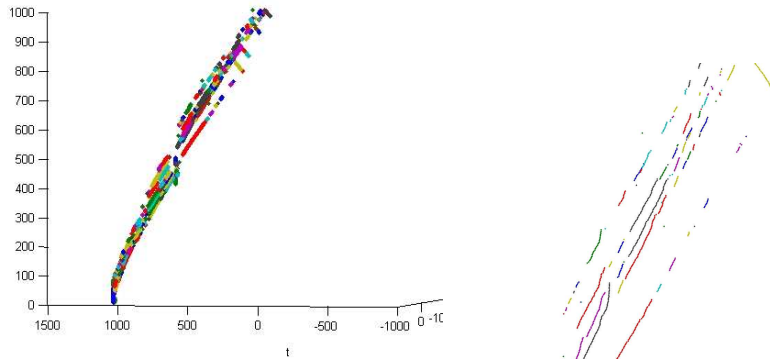


Fig. 8. Tracking trajectories in 3D (left). Zoom in view of Tracking trajectories in 3D(right).

4 Vehicle and person classifier

To classify the detected moving objects into vehicle and person, we design a HOG based vehicle v.s. person classifier for this evaluation task. Our approach can effectively handle multi-view, multi-pose object appearance during the classification. [12]. In our approach, we develop separate classifiers for each object class that are each specialized to one specific aspect or pose. For example, we have one classifier specialized to front/rear view of people and one that is specialized to side view of people. We apply these view-pose-based classifiers in parallel and then combine their results. If there are multiple detections at the same or adjacent locations, the system selects the most likely one through non-maximum suppression.

We empirically determined the number of views/poses to model for each object. Each of these detectors is not only specialized in orientation, but is trained to find the object only at a specified size within a rectangular image window. Therefore, to be able to detect the object at any position within an image, we re-apply the detectors for all possible positions of this rectangular window. Then to be able to detect the object at any size we iteratively resize the input image and re-apply the detectors in the same fashion to each resized image. To build each view-pose-based classifier, we extend the histogram of oriented gradient (HOG) representation and use support vector machines (SVM) as the classifier. Unlike some commonly used representations, the extended histogram of oriented gradient gives good generalization by grouping only perceptually similar images together. With a support vector machine, this gives rise to a decision function that discriminates object and non-object patterns reliably in images under different kinds of conditions and results good performance on some challenging datasets.

A HOG feature is created by first computing the gradient magnitude and orientation at each image sample point in a region around an anchor point. The region is split into $N \times N$ sub-regions. Accumulating samples within the sub-region, weighted by gradient magnitudes, then forms an orientation histogram for each sub-region. Concatenating the histograms from all the sub-regions gives the final HOG feature vector. The standard

Sequence ID	Detection accuracy	Miss detection per frame	False alarm per frame
V4V30002_046	0.000	0.120	0.000
V4V30002_047	0.753	0.724	0.000
V4V30002_048	0.876	0.331	0.005
V4V30002_049	0.929	0.078	0.000
V4V30002_050	0.947	0.107	0.100
V4V30003_011	0.966	0.006	0.067
V4V30003_014	0.622	2.350	0.179
V4V30003_017	0.969	0.039	0.005
V4V30004_005	0.789	0.202	0.157
V4V30003_015	0.868	0.095	0.286
V4V30004_020	0.879	0.107	0.339
V4V30004_021	0.889	0.084	0.241
V4V30004_024	0.923	0.04	0.408
V4V30004_028	0.897	0.022	0.186
V4V30004_029	0.894	0.404	0.061
V4V30004_043	0.945	0.297	0.073
V4V30004_044	0.653	2.293	0.005
V4V30004_046	0.826	0.655	0.293
V4V30004_049	0.960	0.005	0.000

Fig. 9. Part of vehicle tracking results using our algorithms. The most MOTAs are among the range of 0.65 – 0.95. For sequence V4V30002_046, the moving vehicles are only available in a very short period and we did not detect the objects, so the score is 0 with very low false alarm and miss detection.

HOG feature only encodes the gradient orientation of one image patch, no matter where this orientation is from in this patch. Therefore, it is not discriminative enough if the spatial property of the underlying structure of the image patch is crucial. This is especially true for highly structured objects like vehicles. To incorporate the spatial property in HOG feature, we add one distance dimension to the angle dimension in the binning of all the pixels within each sub-region. The distance is relative to the center of each sub-region. We divide the image window into small spatial regions, which consists of a number of sub-regions (or cells). For each cell we accumulate a local 1-D histogram of gradient directions over the pixels of the cell. The combined histogram entries form the representation. For better invariance to illumination, shadowing, etc., it is also useful to contrast-normalize the local responses before using them. This can be done by accumulating a measure of local histogram over somewhat larger spatial regions (or blocks) and using the results to normalize all of the cells in the block.

5 Experimental Results

In this section, we report our systematical tracking results on VIVID2 UAV video sequences for both our tracking algorithms. In the experiment, we test our algorithms on the large-scale VIVID2 data set which contains more than 10 hours videos and 238

video sequences. Each sequence has more than a thousand frames. With the support of NIST evaluation team, 29 video sequences has been selected for person tracking performance evaluation. For the vehicle tracking, we also use the standard Viper tool to generate ground truth for a small data set selected from VIVID2 videos and evaluate our vehicle tracking performance.

In CLEAR evaluation program, a set of performance metrics has been earlier defined for both vehicle and person tracking evaluation[13, 14]. In this paper, we only report three performance metrics from the evaluation program due to the limited space, which include Multiple Object Tracking Accuracy (MOTA), Average Missed Detection per Frame (AMDF), and Average False Alarm per Frame (AFAF). The most important metric MOTA is defined as

$$MOTA = 1 - \frac{\sum_{t=1}^N (c_b(B^t) + c_c(C^t) + \log(Id_{switches}))}{\sum_{t=1}^N (N_G^t)}, \quad (3)$$

where B and C are the false acceptance and false rejection counts, $Id_{switches}$ is total number of identity switches made by the system. Hence, if an identity is changed during the tracking process, it will be penalized by a small number. The perfect tracking result will be given a score equal to 1. The tracking performance become worse when the score becomes smaller or negative. Figure 9 shows the partial vehicle tracking results obtained by using our algorithms, where the most MOTAs are among the range of 0.65 – 0.95 with low false alarm rate. Figure 10 shows the person tracking results. In these result, the MOTAs are quite low due to the small size of moving person and low video quality. Figure 11 shows some frames with detected bounding boxes for both vehicle and person tracking. In the frames, the moving people are always detected by a red bounding box while the moving vehicles are detected by other color bounding box. At the most cases, the moving person has a height of 5-10 pixels and a width of 2-5 pixels. This tiny person is also moving slowly, which creates a serious challenge for the detection stage. In our experiments, we maintain the experimental parameter through all sequences and obtain a stable low false alarm rate, but there are only limited moving person detected in the conducted CLEAR evaluation program.

6 Conclusion

Tracking in UVA video still is a challenging topic in computer vision. In this paper, we present two tracking approaches used in Sarnoff Corporation on a large testing data set. For vehicle tracking, our results demonstrate a reasonable tracking accuracy with low false alarm and missing detection rates. However, for the person tracking, we met a lot of difficulties to achieve good evaluation performance due to the small size of the moving persons, slow motion, and low video contrast. It means a great challenge for person detection and tracking still remained in UAV video. In the future, we will further investigate this challenging case to obtain more stable results for person tracking. We will also integrate the linking process into the online tracking process to improve the tracking performance.

Sequence ID	Detection accuracy	Miss detection per frame	False alarm per frame
V4V30003_042	0.000	0.000	0.000
V4V30002_044	0.000	0.000	0.000
V4V30003_012	0.000	0.000	0.000
V4V30003_013	0.523	0.093	0.109
V4V30004_003	0.000	0.000	0.000
V4V30004_023	0.000	0.000	0.000
V4V30005_030	0.000	0.000	0.000
V4V30007_005	0.030	0.215	0.000
V4V30007_006	0.000	0.000	0.000
V4V30007_016	0.000	0.000	0.000
V4V30007_017	0.000	0.000	0.000
V4V30013_053	0.000	0.000	0.000

Fig. 10. Part of person tracking results using our algorithms.

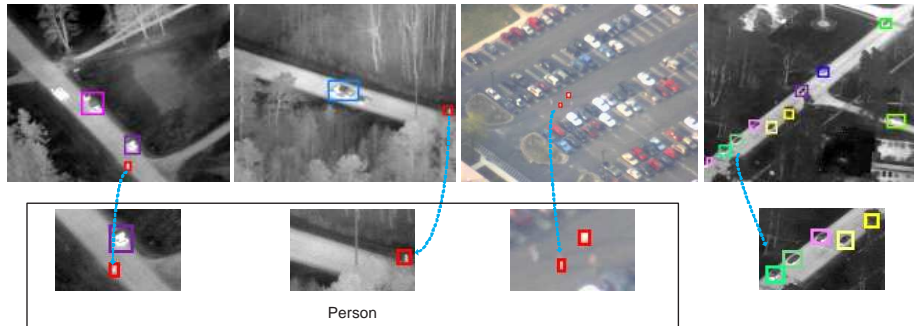


Fig. 11. Some tracking results. The moving people are always detected by a red bounding box, while the moving vehicles are detected by other color bounding box.

References

1. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. *ACM Journal of Computing Surveys* **38** (2006)
2. Shi, J., Tomasi, C.: Good features to track. In: *International Conference on Computer Vision*. (1994) 593–600
3. Tao, H., Sawhney, H., Kumar, R.: Object tracking with bayesian estimation of dynamic layer representations. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **24** (2002) 75–89
4. Yang, C., Duraiswami, R., Davis, L.: Fast multiple object tracking via a hierarchical particle filter. In: *IEEE International Conference on Computer Vision*. (2005) 212–219
5. Perera, A., Srinivas, C., Hoogs, A. Brooksby, G., Hu, W.: Multi-object tracking through simultaneous long occlusions and split-merge conditions. In: *Computer Vision and Pattern Recognition*. (2006)
6. Sawhney, H., Guo, Y., Kumar, R.: Independent motion detection in 3d scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **22** (2000) 1191–1199

7. Kang, J., Cohen, I., Medioni, G., Yuan, C.: Detection and tracking of moving objects from a moving platform in presence of strong parallax. In: IEEE International Conference on Computer Vision. (2005)
8. Xiao, J., Shah, M.: Accurate motion layer segmentation and matting. In: Computer Vision and Pattern Recognition. (2005) 698–703
9. Xiao, J., Shah, M.: Motion layer extraction in the presence of occlusion using graph cuts. IEEE Trans. on Pattern Analysis and Machine Intelligence **27** (2005) 1644–1659
10. Jin, H., Favaro, P., Soatto, S.: Real-time feature tracking and outlier rejection with changes in illumination. In: IEEE International Conference on Computer Vision. (2001)
11. Yang, C., Duraiswami, R., Davis, L.: Efficient spatial-feature tracking via the mean-shift and a new similarity measure. In: IEEE Conference on Computer Vision and Pattern Recognition. (2005)
12. Han, F., Shan, Y., Cekander, R., Sawhney, H., Kumar, R.: A two-stage approach to people and vehicle detection with hog-based svm. In: Performance Metrics for Intelligent Systems Workshop in conjunction with the IEEE Safety, Security, and Rescue Robotics Conference. (2006)
13. Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Manohar, V., Boonstra, M., Korzhova, V.: Performance evaluation protocol for face, person and vehicle detection & tracking in video analysis and content extraction (vace-ii). In: Workshop of Classification of Events, Activities and Relationships. (2006)
14. Manohar, V., Soundararajan, P., Raju, H., Goldgof, D., Kasturi, R., Garofolo, J.: Performance evaluation of object detection and tracking. In: LNCS 3852. (2006) 151–161